

Foundation of Cryptography, Lecture 4

Pseudorandom Functions¹

Handout Mode

Iftach Haitner

Tel Aviv University.

November 20, 2025

¹Last edited on: 2025/12/10.

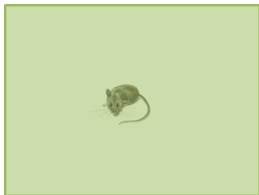
Section 1

Informal Discussion

Motivation discussion

1. We've seen a **small** set of objects: $\{G(x)\}_{x \in \{0,1\}^n}$, that "looks like" a **larger** set of objects: $\{x\}_{x \in \{0,1\}^{2n}}$.
2. We want **small** set of objects: *efficient function families*, that looks like a **huge** set of objects: *the set of all functions*.

Solution



Subsection 1

Function Families

Function families

1. $\mathbb{F} = \{\mathbb{F}_n\}_{n \in \mathbb{N}}$, where $\mathbb{F}_n = \{f: \{0, 1\}^{m(n)} \mapsto \{0, 1\}^{\ell(n)}\}$
2. We write $\mathbb{F} = \{\mathbb{F}_n: \{0, 1\}^{m(n)} \mapsto \{0, 1\}^{\ell(n)}\}$
3. If $m(n) = \ell(n) = n$, we omit it from the notation
4. We identify function with their description

Random functions

Definition 1 (random functions)

For $n, k \in \mathbb{N}$, let $\Pi_{n,k}$ be the family of **all** functions from $\{0, 1\}^n$ to $\{0, 1\}^k$.
Let $\Pi_n = \Pi_{n,n}$.

- ▶ $\pi \leftarrow \Pi_n$ is a “random access” source of randomness
- ▶ Parties with access to a **common** $\pi \leftarrow \Pi_n$ can do a lot
- ▶ How long does it take to describe $\pi \in \Pi_n$? $2^n \cdot n$ bits
- ▶ The truth table of $\pi \leftarrow \Pi_n$ is a uniform string of length $2^n \cdot n$
- ▶ For integer function m , we will consider the function family $\{\Pi_{n,m(n)}\}$.

Subsection 2

Efficient Function Families

Efficient function families

Definition 2 (efficient function family)

An ensemble of function families $\mathcal{F} = \{\mathcal{F}_n\}_{n \in \mathbb{N}}$ is **efficient**, if:

Samplable. \mathcal{F} is samplable in polynomial-time: there exists a PPT that given 1^n , outputs (the description of) a uniform element in \mathcal{F}_n .

Efficient. There exists a polynomial-time algorithm that given $x \in \{0, 1\}^n$ and (a description of) $f \in \mathcal{F}_n$, outputs $f(x)$.

Subsection 3

Pseudorandom Functions

Pseudorandom Functions

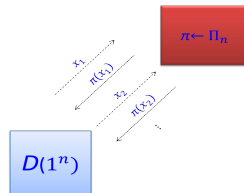
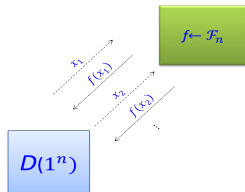
Definition 3 (pseudorandom functions (PRFs))

An efficient ensemble $\mathbb{F} = \{\mathbb{F}_n: \{0, 1\}^{m(n)} \mapsto \{0, 1\}^{\ell(n)}\}$ is **pseudorandom**, if

$$\left| \Pr_{f \leftarrow \mathbb{F}_n} [D^f(1^n) = 1] - \Pr_{\pi \leftarrow \Pi_{m(n), \ell(n)}} [D^\pi(1^n) = 1] \right| = \text{neg}(n),$$

for any oracle-aided PPT D .

\approx_C



- Why “oracle-aided”?
- Easy to construct (no assumption!) with **logarithmic** input length
- PRFs of **super logarithmic** input length, which is the interesting case, imply PRGs
- We will mainly focus on the case $m(n) = \ell(n) = n$
- We write $D^{\mathbb{F}}$ to stand for $(D^f)_{f \leftarrow \mathbb{F}}$.

Section 2

PRF from OWF

Naive Construction

Let $G: \{0, 1\}^n \mapsto \{0, 1\}^{2n}$, and for $s \in \{0, 1\}^n$ define $f_s: \{0, 1\} \mapsto \{0, 1\}^n$ by

- ▶ $f_s(0) = G(s)_{1,\dots,n}$
- ▶ $f_s(1) = G(s)_{n+1,\dots,2n}$.

Claim 4

Assume G is a PRG, then $\{\mathbb{F}_n = \{f_s\}_{s \in \{0,1\}^n}\}_{n \in \mathbb{N}}$ is a PRF.

Proof: The truth table of $f \leftarrow \mathbb{F}_n$ is $G(U_n)$, where the truth table of $\pi \leftarrow \Pi_{1,n}$ is U_{2n} □

- ▶ Naturally extends to input of length $O(\log n)$:-)
- ▶ Miserably fails for longer length (which is the only interesting case) :-)
- ▶ Problem, we are constructing the **whole** truth table, even to compute a **single** output

Subsection 1

The GGM Construction

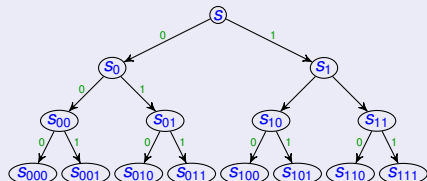
The GGM Construction

Construction 5 (GGM)

For $G: \{0, 1\}^n \mapsto \{0, 1\}^{2n}$ and $s \in \{0, 1\}^n$,

- ▶ $G_0(s) = G(s)_{1,\dots,n}$
- ▶ $G_1(s) = G(s)_{n+1,\dots,2n}$

For $x \in \{0, 1\}^k$ let $f_s(x) = G_{x_k}(f_s(x_1, \dots, x_{k-1}))$,
letting $f_s() = s$.



$$s_x = f_s(x)$$

- ▶ Example: $f_s(001) = s_{001} = G_1(s_{00}) = G_1(G_0(s_0)) = G_1(G_0(G_0(s)))$
- ▶ G is poly-time $\implies \mathbb{F} := \{\mathbb{F}_n = \{f_s: s \in \{0, 1\}^n\}\}$ is efficient

Theorem 6 (Goldreich-Goldwasser-Micali (GGM))

If G is a PRG then \mathbb{F} is a PRF.

Corollary 7

OWFs imply PRFs.

Subsection 2

Proof

Proof Idea

Assume \exists PPT D , $p \in \text{poly}$ and infinite set $\mathcal{I} \subseteq \mathbb{N}$ with

$$|\Pr[D^{\mathbb{F}_n}(1^n) = 1] - \Pr[D^{\Pi_n}(1^n) = 1]| \geq \frac{1}{p(n)}, \quad (1)$$

for any $n \in \mathcal{I}$.

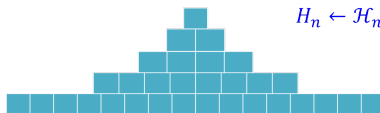
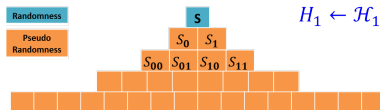
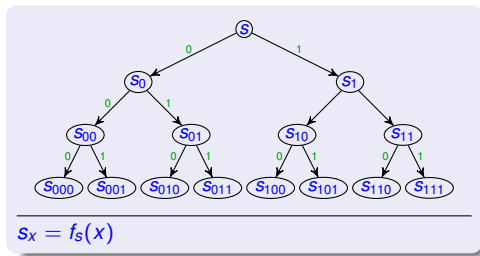
Fix $n \in \mathbb{N}$ and let $t = t(n)$ be a bound on the running time of $D(1^n)$. We use D to construct a PPT D' such that

$$|\Pr[D'((U_{2n})^t) = 1] - \Pr[D'(G(U_n))^t = 1]| > \frac{1}{np(n)},$$

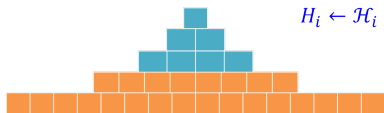
where $(U_{2n})^t = U_{2n}^{(1)}, \dots, U_{2n}^{(t)}$ and $G(U_n)^t = G(U_n^{(1)}), \dots, G(U_n^{(t)})$.

Hence, D' violates the security of G .(?)

The Hybrid



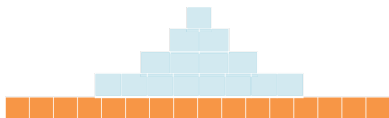
- ▶ \mathcal{H}_i : all the nodes of depth smaller equal to i are labeled by random strings. Other nodes are labeled as before (by applying PRG to the father and taking right/left half).
- ▶ What family is \mathcal{H}_1 ? \mathbb{F}_n . What is \mathcal{H}_{n+1} ? Π_n .
- ▶ For some $i \in \{1, \dots, n-1\}$, algorithm D distinguishes \mathcal{H}_i from \mathcal{H}_{i+1} by $\frac{1}{np(n)}$



$\not\approx$

The Hybrid cont.

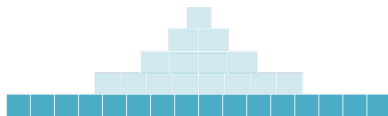
We focus on the case where D distinguishes between \mathcal{H}_{n-1} and \mathcal{H}_n



- ▶ D distinguishes (via t samples) between
 - ▶ R – a uniform string of length $2^n \cdot n$, and
 - ▶ P – a string generated by 2^{n-1} independent calls to G
- ▶ We would like to use D for breaking the security of G , but R and P seem too long :-)
- ▶ Solution: focus on the part (i.e., cells) that D sees

The Hybrid cont.

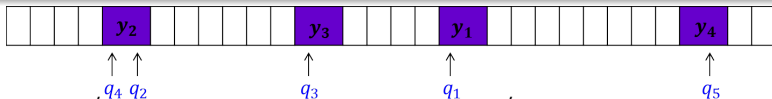
We focus on the case where D distinguishes between \mathcal{H}_n and \mathcal{H}_{n+1}



Algorithm 8 (D' on $y_1, \dots, y_t \in (\{0, 1\}^{2n})^t$)

Emulate D . Initialize a counter $k = 0$. On the i 'th query q_i made by D :

- ▶ If the cell queried by q_i is **non-empty**, answer with the content of the cell.
- ▶ Else increment k by 1 and do:
 - ▶ If q_i is a left son, fill its cell with the left half of y_k and use the right half of y to fill the right brother of q_i .
 - ▶ If q_i is a right son, fill its cell with the right half of y_k and use the left half of y to fill the cell of left brother of q_i .



- ▶ $D'(U_{2n})^t / D'(G(U_n))^t$ emulates D with access to R / P
- ▶ Hence, $|\Pr[D'((U_{2n})^t) = 1] - \Pr[D'(G(U_n))^t) = 1]| > \frac{1}{np(n)}$

Part I

Pseudorandom Permutations

Formal definition

Let $\tilde{\Pi}_n$ be the set of all permutations over $\{0, 1\}^n$.

Definition 9 (pseudorandom permutations (PRPs))

A permutation ensemble $\mathbb{F} = \{\mathbb{F}_n : \{0, 1\}^n \mapsto \{0, 1\}^n\}$ is a **pseudorandom permutation**, if

$$\left| \Pr[\mathbb{D}^{\mathbb{F}_n}(1^n) = 1] - \Pr[\mathbb{D}^{\tilde{\Pi}_n}(1^n) = 1] \right| = \text{neg}(n), \quad (2)$$

for any oracle-aided PPT \mathbb{D}

- ▶ Eq 2 holds for any PRF (taking the role of \mathbb{F})
- ▶ Hence, PRPs are indistinguishable from PRFs...
- ▶ If no one can distinguish between PRFs and PRPs, let's use PRFs
 - ▶ (partial) Perfect "security"
 - ▶ Inversion

Subsection 1

PRP from PRF

Feistel permutation

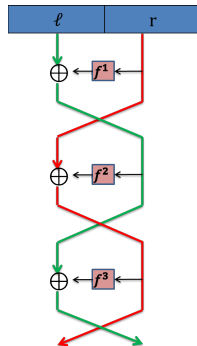
How does one turn a function into a permutation?

Definition 10 (LR)

For $f: \{0, 1\}^n \mapsto \{0, 1\}^n$, let $\text{LR}_f: \{0, 1\}^{2n} \mapsto \{0, 1\}^{2n}$ be defined by

$$\text{LR}_f(\ell, r) = (r, f(r) \oplus \ell).$$

- ▶ LR_f is a permutation: $\text{LR}_f^{-1}(z, w) = (f(z) \oplus w, z)$
- ▶ LR_f is **efficiently** computable and invertible given oracle access to f
- ▶ For $i \in \mathbb{N}$ and f^1, \dots, f^i , define $\text{LR}_{f^1, \dots, f^i}: \{0, 1\}^{2n} \mapsto \{0, 1\}^{2n}$ by
$$\text{LR}_{f^1, \dots, f^i}(\ell, r) = (r^{i-1}, f^i(r^{i-1}) \oplus \ell^{i-1}), \text{ for } (\ell^{i-1}, r^{i-1}) = \text{LR}_{f^1, \dots, f^{i-1}}(\ell, r).$$
(letting $(\ell^0, r^0) = (\ell, r)$)



Luby-Rackoff Thm.

Recall $\text{LR}_f(\ell, r) = (r, f(r) \oplus \ell)$.

Definition 11

Given a function family $\mathbb{F} = \{\mathbb{F}_n: \{0, 1\}^n \mapsto \{0, 1\}^n\}$, let

$$\text{LR}^i(\mathbb{F}) = \{\text{LR}_{\mathbb{F}_n}^i = \{\text{LR}_{f^1, \dots, f^i}: f^1, \dots, f^i \in \mathbb{F}_n\}\},$$

- ▶ $\text{LR}_{\mathbb{F}}^i$ is always a permutation family, and is efficient if \mathbb{F} is.
- ▶ Is $\text{LR}_{\mathbb{F}}^1$ pseudorandom?
- ▶ $\text{LR}_{\mathbb{F}}^2$? $\text{LR}_{f^1, f^2}(0^n, 0^n) = \text{LR}_{f^2}(0^n, f^1(0^n)) = (f^1(0^n), \cdot)$
and $\text{LR}_{f^1, f^2}(1^n, 0^n) = \text{LR}_{f^2}(0^n, f^1(0^n) \oplus 1^n) = (f^1(0^n) \oplus 1^n, \cdot)$
- ▶ $\text{LR}_{\mathbb{F}}^3$?

Theorem 12 (Luby-Rackoff)

Assuming that \mathbb{F} is a PRF, then $\text{LR}_{\mathbb{F}}^3$ is a PRP

- ▶ $\text{LR}_{\mathbb{F}}^4(\mathbb{F})$ is pseudorandom even if **inversion queries** are allowed

Proving Luby-Rackoff

It suffices to prove that $\text{LR}_{\Pi_n}^3$ is pseudorandom (?)

- ▶ How would you prove that?
- ▶ Maybe $\text{LR}^3(\Pi_n) \equiv \tilde{\Pi}_{2n}$? description length of element in $\text{LR}^3(\Pi_n)$ is $2^n \cdot 3n$, where that of element in $\tilde{\Pi}_{2n}$ is $\log(2^{2n}!) > \log\left(\left(\frac{2^{2n}}{e}\right)^{2^{2n}}\right) > 2^{2n} \cdot n$

Claim 13

For any q -query D ,

$$|\Pr[D^{\text{LR}^3(\Pi_n)}(1^n) = 1] - \Pr[D^{\tilde{\Pi}_{2n}}(1^n) = 1]| \in O(q^2/2^n).$$

- ▶ We assume for simplicity that D is *deterministic*, *non-repeating* and *non-adaptive*.
- ▶ Let x_1, \dots, x_q be D 's queries.
- ▶ We show $\{f(x_i)\}_{f \leftarrow \text{LR}^3(\Pi_n)}$ is $O(q^2/2^n)$ close (i.e., in statistical distance) to $(U_{2n})^q$. Is that enough?

$(f(x_0), \dots, f(x_q))_{f \leftarrow \text{LR}^3(\Pi_n)}$ is close to $(U_{2n})^q$

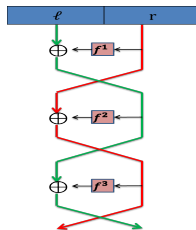
Let $(\ell_1^0, r_1^0), \dots, (\ell_q^0, r_q^0) = (x_1, \dots, x_q)$.

The following rv's are defined w.r.t. $(f^1, f^2, f^3) \leftarrow \Pi_n^3$.

ℓ_1^0	r_1^0	ℓ_2^0	r_2^0	...	ℓ_q^0	r_q^0
ℓ_1^1	r_1^1	ℓ_2^1	r_2^1	...	ℓ_q^1	r_q^1
ℓ_1^2	r_1^2	ℓ_2^2	r_2^2	...	ℓ_q^2	r_q^2
ℓ_1^3	r_1^3	ℓ_2^3	r_2^3	...	ℓ_q^3	r_q^3

where $\ell_b^j = r_b^{j-1}$ and $r_b^j = f^j(r_b^{j-1}) \oplus \ell_b^{j-1}$.

Let $\text{Bad}^c = \{\exists(i, c) \neq (j, b \leq c): r_i^c = r_j^b\}$.



Claim 14

$$\Pr[\text{Bad}^1] \leq \frac{q^2}{2^n} \text{ and } \Pr[\text{Bad}^2] \leq 3 \cdot \frac{q^2}{2^n}$$

Proof: $r_i^0 = r_j^0 \implies r_i^1 \neq r_j^1 \dots$

Claim 15

$$(1) \{r_i^2\}_{i \in \neg \text{Bad}^1} \equiv U_n^q \quad (2) \{r_i^3\}_{i \in \neg \text{Bad}^2, \{r_i^2\}} \equiv U_n^q$$

Section 3

Applications

General paradigm

Design a scheme assuming that you have random functions, and the **realize** them using PRFs.

Subsection 1

Private-key Encryption

Private-key Encryption

Construction 16 (PRF-based encryption)

Given an (efficient) PRF \mathbb{F} , define the encryption scheme (Gen, E, D) :

Key generation: $\text{Gen}(1^n)$ returns $f \leftarrow \mathbb{F}_n$

Encryption: $E_f(m)$ returns $(U_n, f(U_n) \oplus m)$

Decryption: $D_f(c = (c_1, c_2))$ returns $f(c_1) \oplus c_2$

- ▶ Advantages over the PRG based scheme?
- ▶ Proof of security?

Conclusion

- ▶ We constructed PRFs and PRPs from length-doubling PRG (and thus from one-way functions)
- ▶ Main question: find a simpler, more efficient construction or at least, a less **adaptive** one